Total Ship Systems Engineering

Mr. James D. Horner

There is much need for defining a framework for systems engineering of Navy ships. This framework must embody the current Navy goals for developing cost-effective systems. In particular, new approaches for systems development need to be defined based on acquisition reform principles. An important aspect of any new ship development approach is defining the development process in terms of proper roles and responsibilities.

Future ship development will require a greater amount of systems analysis that includes ship modeling, simulation, and prototyping. The analysis needs to support a continuous process of refining ship systems requirements. Proper structuring of ship requirements and proper application of analysis methodologies is key to developing concise, complete specifications. Conducting effective trade studies requires a process whereby prototyping can work in conjunction with ship modeling.

Powerful analysis tools and methods exist that can be used to define ships at all levels: mission, operational and architectural. Operational process analysis is key in determining effective shipboard battle management organizations (BMOs). Activity-based costing (ABC) is readily combined with these analysis methods to produce ship models to support cost-performance trade studies.

Ship integration will require better integration at the program and engineering level. Future engineering development environments must provide an integrating framework of tools and methods to support analysis, design, and trade studies. New mission areas are requiring integration and reengineering of ship systems across traditional warfare areas. Tracking the development of the various ship systems requires integration of the program data management environments.

Introduction

The key to the success of any program or project is effective control and leadership. For ship development, this requires a program-level control based on a well-defined systems engineering framework. This framework should specify the systems engineering functions required for effective management. Indeed, this process itself of defining and getting concurrence on the systems engineering framework is where effective control and leadership begins.

A well-defined systems engineering framework makes clear how the work is to be completed and the various efforts coordinated. Most importantly, a systems engineering framework forms a real basis for assigning roles and clarifying their relationships.

A systems engineering framework also must define the system engineering functions and processes in terms of supporting program office responsibilities (monitoring, evaluating, and planning). Interfaces between systems engineering and program office functions need to be identified in terms of reporting, recommendations, approvals, and direction. It should also specify the processes for supporting development of program plans and work statements. In this way, the process descriptions establish the system engineering responsibilities and organizational role within a program.

Defining this Total Ship Systems Engineering (TSSE) framework begins with specifying ship development models that encompass the entire life cycle of the ship. It must identify the important systems engineering concepts and define the most relevant processes at each life-cycle phase. These become the basis for establishing a common understanding of the development process throughout the Navy.

A well-grounded understanding of the basic functions of systems engineering is essential to defining this framework. However, the TSSE framework must also reflect the particulars of our current Navy ship development environment. This includes mandates and guidance arising from acquisition reform and cost as an independent variable (CAIV), along with current Navy goals such as reduced manning and total ownership cost (TOC).

TSSE AND ACQUISITION REFORM

A major impact of acquisition reform, with respect to systems engineering, is that ship requirements are continually refined during development. This is in contrast to having complete, detailed specifications established prior to design. A process of continual refinement of ship requirements can have a positive impact on ship development for a number of reasons.

First, the sheer complexity of Navy ships makes it nearly impossible to develop both detailed and justifiable specifications (from a cost-performance viewpoint) without the support of considerable modeling, simulation, prototyping and other analysis efforts. Much of these analysis and trade studies can, and arguably should, take place during architecture development and design.

Second, considerable cost savings may be achieved by having ship specifications reflective of those resources; such as the facilities, legacy systems, and people skills that are or could be made available to a primary contractor. That is, it is cost effective to have requirements leverage the unique capabilities and resources of the development team.

Third, the ship must include the shipboard personnel as part of the total ship system in order to maximize overall performance to cost. That is, manning and hardware (HW)–software (SW) systems must both remain part of the trade space as ship functionality is defined, and the cost and performance estimates of those functions are refined.

Fourth and perhaps most importantly, with today's emphasis on total life-cycle cost (LCC), there is the important need to address ownership cost issues. Ship system quality factors such as maintainability and upgradability will obviously impact ownership costs. Ownership costs also include operator training costs. New ship systems can also impose huge development and upgrade costs on systems external to the ship. These issues are difficult, if not impossible, to adequately address in a predesign, performance-type specification.

Thus, there can be many benefits in continuing requirements definition during the early stages of ship design. It provides an opportunity for specifying ship requirements that represent better value to the customer. However, this process can work only if the customer maintains control of the requirements. The government must still perform whatever types of evaluations or however much analysis is necessary to specify the systems being procured. This is the government's obligation to the taxpayer.

Unfortunately, acquisition reform is often misinterpreted as implying that the government must give up control of specifying the requirements. When ever a system is procured using either vague or unverifiable requirements, control is being relinquished. Worse yet, many of the activities that are necessary to properly specify requirements are no longer being properly supported. These include requirements analysis, functional analysis, and supporting trade studies.

Adding to the confusion is the distinction being made between *what* versus *how* in the specifying of systems. As any designer knows, *how* is always just the next level of detail for *what*. That is, as the system definition is refined, each level of *how* becomes the *what* for the next level of detail. Different systems require different levels of specification detail depending on a number of factors.

Attempting to differentiate between the two can result in increased spending without requiring anything specific in return. This also damages instituting a systems engineering approach for military systems development.

Operational requirements documents or mission needs statements documents are necessary for creating a context for requirements analysis. However, requirements become verifiable only at a much greater level of detail. In addition, the government must assess system component quality issues as they impact life-cycle operational and maintenance costs. For maximum value to the customer, every type of ship component that is to be developed or purchased should be given the same attention it would receive if it were being separately procured.

If the customer depends on the shipbuilder to generate verifiable requirements, it can result in long-term dependence for expensive maintenance of very custom or company-proprietary ship components. In addition, builder-generated specifications inevitably impose, or in many cases completely defer, requirements onto other customer-owned systems and activities.

TSSE MODEL

Figure 1 introduces a notional TSSE model for the initial phases of ship development. The model focuses on systems engineering and development from the customer's viewpoint. It can be viewed as a concept for developing a systems engineering-based management plan. It should be noted that each of the major functional areas might in themselves require development of one or more plans.

The program and project level functions necessary to define ship requirements and refine them into specifications are identified in Figure 1. It also identifies the **Integration** elements required to support these functions. A third area labeled **Control** identifies those core management functions necessary to ensure that the ship is properly specified and built according to those specifications.

Control

The **Control** function of Figure 1 defines the processes whereby requirements are defined. This involves establishing, maintaining, and enforcing the processes by which the ***Requirements & Functional Analysis*** and ***Prototyping & Trade Studies*** are conducted and interact.

The **Control** function must define the project and program environments necessary to support these activities. Environment definition includes the tools, data/information support systems, and the processes by which they are used. The development environment must ensure proper dissemination of program- and project-level information at all phases of development. This is referred to in Figure 1 as **Integration** because these environments identified are critical to maintaining project and program integration.

Control is the key in procuring or developing a product of good value. Maintaining control of the development of large-scale complex systems requires strong and competent program systems engineering support. Deficiencies in systems engineering support results in wasted money, time, and other resources.

Unfortunately, program managers often have greater incentives to deliver at cost and on time than they have to deliver quality products. That is, there is an

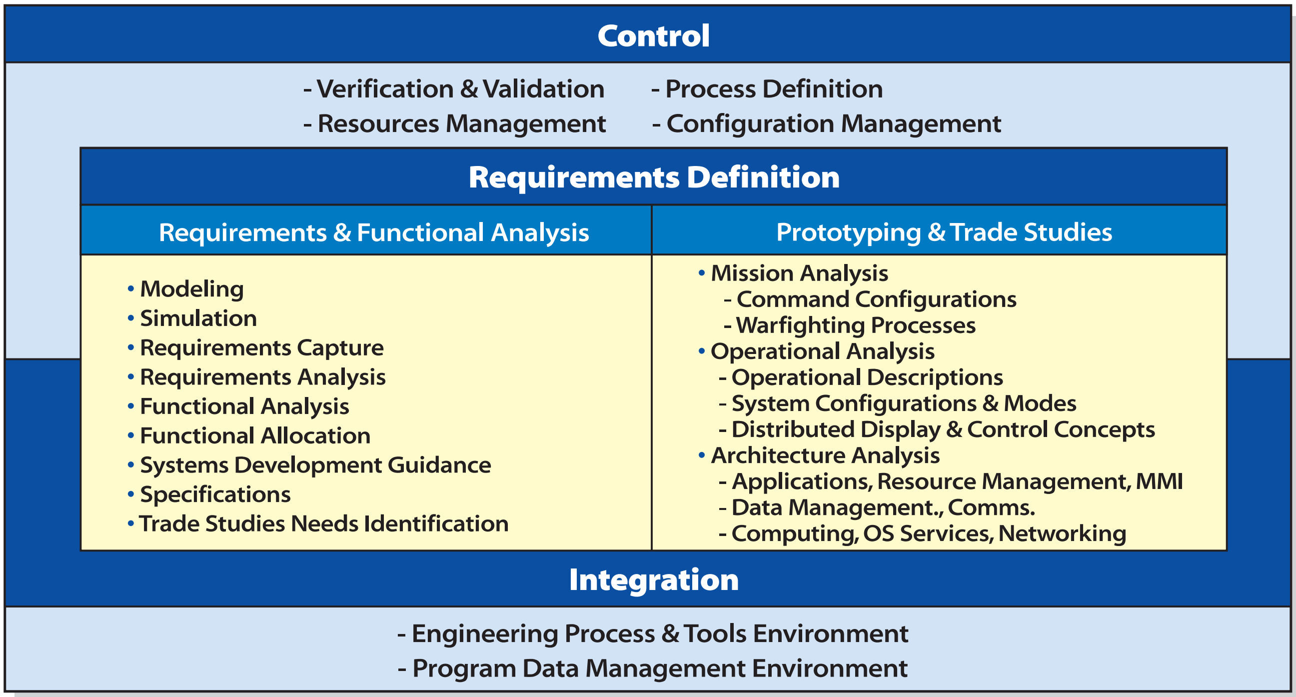

Figure 1—TSSE Model (Customer View)

emphasis on delivering "something" at cost and on time. One reason for this is that cost and schedule are easily quantifiable, whereas quality and value are not. In addition, the quality of the system and other deliverables cannot be readily discerned during development.

Because of the relative lack of incentives regarding the quality of deliverables, system specification and verification functions are often not viewed as program risk-reduction activities. Indeed, these functions may be considered as only adding "risk" to program cost and schedule. The net effect is that a customer-based systems engineering role is almost never formally established within a development program.

This tends to benefit the contractors but certainly provides little benefit to the government. It is a primary reason today for exorbitant costs and for systems being delivered that do not perform properly. Unfortunately, acquisition reform is often interpreted as supporting this management concept. That is, systems engineering is not understood as the essential customer function for effective control and management.

Establishing competition within the acquisition process does assist in ensuring delivery of quality products. However, it is only one of the factors. No amount of competition can replace the customer's need to be a smart buyer. This implies the need for a development process whereby the customer learns as the acquisition proceeds through its phases. For the development of complex systems such as ships, knowledge is obtained through systems analysis. That is, systems analysis provides the knowledge base required for effective program monitoring, assessment, and planning.

The government must better institute within itself incentives for delivering quality products, both final and interim. It is only then that the necessary customer-sided systems engineering roles will be established. This applies not only to the program offices, but to the research and development centers as well. However, the systems engineering role should include responsibilities for cost and schedule as well as system performance. That is, there needs to be a sharing of responsibilities between program management and system engineering functions.

Building Industry Comparison

Examination of system engineering issues in industries other than the Department of Defense (DOD) can be useful in order to gain some perspective. In the example of the commercial building industry, there are three primary roles in construction: the customer, the architect, and the builder. The architect role is very similar to that of the systems engineer.

The architect must determine the customer's wants and needs and translate them into blueprints and specifications. Throughout development, the architect assists the customer in ensuring that the building is constructed as specified (verification). The quality of the specifications is determined by the amount of detail. Specifications should include a list of materials, with the quality of all materials specified.

The specifications are then put out on competitive bid. The builders at this point have an opportunity to make changes to the specifications. Changes are typically desired in order to take advantage of their subcontractor's particular expertise or to leverage materials that may already be on hand. The architect assists the customer in negotiating specification changes with the builder anytime during the project.

There are many lessons that can be learned from the construction and building industry. In the construction industry, lessons do not tend to get lost between major projects. For instance, it is known that the competitive bid process can only be a supplement to, not a replacement of, a complete and detailed set of specifications. The importance of knowledgeable and continuous inspection or verification is also known.

However, establishing proper roles is viewed as the most critical aspect of a successful project. It is well known in the construction industry that a single corporation should never provide both the architect and building services. There is no other arrangement that can better guarantee project disaster. Indeed, the architect is ethically and legally bound to represent the interests of the customer.

Personal Computer (PC) Industry Comparison

Consolidation of the defense industry continues to bring rapid changes to the Navy. Some of the most profound changes are a consequence of the aging workforce for both government and industry alike. The defense industry is losing (proportionately) much of its technical skill base. Whether consolidation continues or not, work efficiencies will be negatively impacted for decades to come.

There is much pressure within the private sector to continue to provide increasing return on investment for shareholders and upward mobility for employees. For the large defense corporations, this is probably not achievable as long as their primary business function is to provide highly technical design and building services. Their workforce makeup puts them at a competitive disadvantage with the smaller engineering firms.

The response of the large defense corporations, in terms of competitive repositioning, appears to be following a strategy similar to that adopted by Microsoft. Microsoft's strategy was to reorient towards a more marketing focus, backed with some systems integration expertise. Smaller, more agile companies today provide a much greater percentage of the design and development for Microsoft products.

However, there needs to be concerns with the large defense company's new focus on marketing and integration. That problem is related to the differences between military systems and PCs and what is required for their developments. Real-time and/or fault-tolerant computing requires exacting systems design and a high degree of integration. Meeting requirements for real-time and fault-tolerant computing requires a total system perspective. Conversely, PC SW and HW is not engineered as a total system. In the PC industry, there are no total system requirements regarding performance or architecture. Thus, PCs are notoriously buggy and (relatively) poorly integrated.

Ship systems are being required to achieve a higher degree of integration. This is the result of the new joint mission areas such as **area defense** and **land attack**. A greater number of ship systems must become involved in traditionally "mission critical" functions such as tactical planning and weapon targeting. This will require that a greater percentage of ship control systems have real-time, fault-tolerant computing capabilities. As systems become more integrated, the mission-critical systems can impose requirements on other systems for real-time, fault-tolerant computing.

Thus ship systems will, in the future, migrate toward mission-critical application and become more integrated each other. This describes a very large-scale, highly integrated computing system. It also describes a systems design problem that requires exacting analysis at the total-systems level. This has little or no relation to the "plug and play" integration problem for PCs or PC-based networks. The government must maintain the capabilities for performing these types of analysis.

Requirements & Functional Analysis

The ***Requirements & Functional Analysis*** efforts of Figure 1 should be managed as a continuous process of requirements refinement. That is, functional analysis, supported by trade studies, should be viewed as the primary means of refining requirements into concise specifications.

However, requirements need to be well structured, even before functional analysis begins. Requirement types need to be identified in order to get an understanding those aspects of the systems that should be specified. Structuring requirements also aids in understanding the level of detail needed. Typing and structuring of requirements is important regardless of how the requirements are to be eventually organized.

There are available methodologies and tools for performing functional analysis. Structured analysis methods support functional analysis based on system modeling. The models are able to break down a system's complexity through two different means. First, the analysis models partition a system along its dimensions of functional, data, and control. Second, the models support top-down decomposition of system functions and data.

A structured modeling approach to functional analysis can also be used in conjunction with trade studies for specifying a system. The functional models can provide a means for more readily capturing the results of the trade studies. Having the requirements well organized and structured also assists in identifying the needed trade studies. That is, the analysis models make explicit the "holes" in the requirements and thus help establish the metrics for trade studies.

Requirements Analysis

Requirements analysis continues to that level of detail necessary to ensure that the customer knows exactly what is desired. Development of the specifications continues until the customer is confident of getting the desired product. Neglecting requirements analysis can result in unexpected, and in some cases exorbitant, development and/or ownership costs. Another consequence is an inadequate skill or knowledge base for performing these types of analysis. Unfortunately, there are strong incentives for neglecting requirements analysis. These include the desire to show immediate progress and to minimize up-front development costs.

In Table 1, a decomposition of the requirements refinement (***Requirements & Functional Analysis***) process of Figure 1 is presented. In this model, ship requirements are viewed as coming from three distinct considerations: mission, operations, and architecture. Operations analysis is important for gaining an understanding of operational costs and requirements for system flexibility. Similarly,

Table 1—Requirements and Functional Analysis Model

	MISSION REQUIREMENTS	OPERATIONAL REQUIREMENTS	ARCHITECTURAL REQUIREMENTS
PHASE I BASELINING	✥ Scenarios Definitions ✥ Threat Descriptions	✥ Baseline Process Modeling ✥ Operational Principles	✥ System Performance Baseline Capture
PHASE II CONTEXT DEFINITION	✥ External Event Lists ✥ Environmental Conditions	✥ Operations Modeling ✥ Distributed Control Concepts	✥ Systems Architectural Principles ✥ Maintenance & Upgrade Approaches
PHASE III REQUIREMENTS REFINEMENT	✥ Warfighting Processes ✥ Command Configurations	✥ Operations Descriptions ✥ System Configurations and Modes ✥ Training Requirements	✥ System Architecture ✥ Maintenance and Upgrade Approaches

architectural analysis is important in estimating the costs associated with anticipated system maintenance and upgrades.

Requirements can also be classified into types, which for defense systems, may best be partitioned into behavioral (functional and timing) requirements, interface requirements, and environmental requirements. Taking these three requirement types across the above three requirement sources provides a structure for performing requirements analysis. This structure helps ensure that specifications are established for exactly those areas that need specification.

Mission Requirements—One of the important aspects of establishing mission requirements is that, whenever possible, they are stated in terms that do not imply a SW, HW, or "humanware" type solution (a particular implementation). For example, total ship operational availability (Ao) should be defined as total system Ao, including the operators. Similarly, ship survivability should possibly include key shipboard personnel (safety) in its definition.

At the mission level, the ship should first be viewed as a black box, with emphasis given on defining its total environment and how it interfaces and interacts within its environment. In structured analysis, this represents the context level where external system interfaces are defined, and its behavior in totality is described in terms of responding to defined external events. For Navy ships, the external events are typically generated through analysis of the mission scenario descriptions in conjunction with threat profiles.

This ship context description forms a basis for analyzing how the ship can be effectively fought during various missions. Alternate command configurations are evaluated in terms of meeting various mission objectives. This type of analysis is also useful for refining alternative off-ship command and control structures and, thus, for refining theater-level warfighting strategies.

Models should be developed that describe the propagation of control through the alternate command structures to the major warfighting assets, including the ship itself. The ***Prototyping & Trade Studies*** activities support mission analysis through simulations involving mock-ups of ship command and control centers, including command positions. These simulations also provide an important means of obtaining feedback from fleet personnel.

The analysis of these simulations should focus on evaluating warfighting processes against variations in mission scenarios. Analysis tools should be used for capturing and refining the results from the warfighting simulations. Warfighting processes include descriptions of the command structures, their communications, major ship assets, and their control. Together, these describe the mission-level requirements for the ship.

Operational Requirements—Proper specifications of ship operational requirements are important for a number of reasons. First, it begins to formalize those aspects of the ship and supporting Navy infrastructure that are to be considered for change (part of the trade space) and those that are not (a system constraint). Second, they are much more verifiable than mission-level requirements. Third, it is only through operations modeling that the requirements for system flexibility and reconfiguration can be identified.

Operational analysis refines the command structures and high-level warfighting processes. Each command position identified through mission analysis is analyzed to determine the type of support that is required. Support could come from direct access to information sources, operator manning of consoles, or command decision aids. That is, resources are allocated to the warfighting process according to their effectiveness in supporting command. Other considerations include LCCs and flexibility. It is at this level that human versus machine allocations are initially defined.

Based on this initial allocation, high-level warfighting processes are refined into shipboard operational processes. Shipboard machine functions and human tasks are identified. Activity analysis is used to identify optimal operational processes. If scenario timelines cannot be met without overloading resources, then:

- Additional machines, equipment or personnel need to be added, or
- Operational processes need to be further streamlined. The high-level functional descriptions combined with timing constraints form the shipboard operational process models.

One crucial area of operational interface-type requirements is the man-machine interfaces (MMIs). In addition to representing major system components, the MMIs have great impact on overall ship performance. There should be a continual refinement of shipboard display and control concepts. Development of these concepts need not wait until machine versus personnel allocations have been finalized.

Architectural Requirements—Today, concepts for new acquisition processes that avoid ownership cost risk issues are being explored. One such concept is that of single-contractor awards for both development and life-cycle maintenance. However, operational and maintenance costs are inherently ownership problems. As such, their risks cannot be readily shifted onto the seller. The only effective means of controlling ownership costs is through smart acquisitions.

Operational requirements (processes) infer requirements on the shipboard HW and SW systems. The ship's systems must provide the functionality and performance required to support all command and operator positions. Operational requirements also infer architectural requirements on a system for reconfigurability. Systems must support various operational modes that reflect alternate command configurations and supporting operational processes. Additional system flexibility is typically desirable for supporting undefined future missions.

However, there is another important class of architecture requirements that does not flow down from the operational requirements. This class of requirements addresses issues of system maintainability and upgradability. It is through consideration of ship and ship systems architecture that quality attributes of systems can be adequately specified. The systems architecture and system components quality attributes can have a tremendous impact on the costs to maintain and upgrade the systems.

Consideration must also be given to the kinds of support facilities and resources that are currently available in the Navy to maintain the systems. Required changes to Navy ship support infrastructure could impose costs that may take many years to recoup. Although the bottom line is assumed to be total LCCs, most managers also view the payback period as an important metric. This is because a long payback period can be risky in terms of actually realizing the savings.

The operational requirements impose functional requirements on the applications and the human-computer-interface SW. These, in turn, impose requirements on the data management and communications “middleware.” Middleware support imposes requirements on machine operating systems (OS). Finally, the machine OS imposes requirements on the machines themselves and other HW.

At each level, quality attributes of the system components must be specified that cannot be directly inferred by the higher level requirements. Attributes such as Ao or reliability can indeed be specified at the ship level. In addition, these measures can be allocated across ship systems and components. However, such measures can be met in a number of ways depending on the types and amount of ship systems maintenance that is assumed. It is for this reason that quality attributes must be specified that take into consideration ownership costs.

Today there is considerable emphasis on utilizing commercial off-the-shelf (COTS) components with open-system interfaces. Obviously, a COTS solution cannot always make sense based on performance, LCC trade-offs and risks. One major problem in using COTS in military systems is verification. No amount of black box testing can indicate that a component does not have some “extra” function. It is only through examining the component design or source code that it can be verified that a component will not do what it is not supposed to do. In any case, issues regarding the selection, use, and testing of COTS components needs to be better addressed by DOD.

Modeling-Based Functional Analysis

An important aspect of defining the TSSE framework is specifying the kinds of analysis, along with their associated tools and methods, that are to be performed at each phase of ship development. Structured analysis provides a means of performing functional analysis through modeling. It is often referred to as *essential modeling* because the models produced are suppose to capture the “essence” of what the system must do.

Structured Analysis produces functional models that are concisely defined. Indeed, it is only well-defined entities or elements that can meaningfully be assigned performance and cost attributes. For example, functional models produced by Structured Analysis have a well-defined syntax for defining their data flows and utilize state transition logic for modeling of functional timing and control.

Structured Analysis models can represent a system at multiple levels of generalization or abstraction without being vague or ambiguous. In addition, the models produced through Structured Analysis are also, by definition, implementation independent. Thus, Structured Analysis provides a powerful means of analyzing system functional requirements while avoiding design issues.

Through Structured Analysis, functional requirements are analyzed through system decomposition and partitioning. These are the two primary means of breaking down the complexities of systems. Partitioning occurs along the system’s dimensions of data, control/timing, and functions/activities. Methods exist for system decomposition along each of these dimensions.

Process Analysis—An important recognition that occurred as a result of acquisition reform, is that the ship and its systems should be viewed as including the human operators. Traditionally, the customer side of the Navy predetermined the shipboard BMO. A system-level specification was then generated from the BMO according to the perceived need for HW and SW support.

Today, the human operator must be part of the cost-performance analysis trade-space. Traditional HW and SW tools and methods used in structured analysis are no longer adequate for analyzing ships and ship systems. Process analysis methods must also be utilized to model shipboard activities that may include operator functions (see the Operational Requirements of Table 1).

Process engineering creates a context for systems engineering of HW/SW components and human engineering. This is similar to how systems engineering provides a framework within which SW and HW engineering are performed. The business process reengineering (BPR) industry has tools and structured methods available that are well suited for shipboard operational process modeling.

The functional, data, and control elements of process models may best be characterized as abstractions of those used in Structured Analysis for HW/SW modeling. They are abstracted so as to allow for the possibility that system components may be of the human or machine type. For example, instead of using logical control for modeling timing, it may use a more general construct such as "constraint." A constraint may be anything from a user's manual to real-time operational command guidance. Process modeling typically allows for input/output types other than data such as fuel or parts for maintenance.

The importance of utilizing process analysis tools and methods for ship operational modeling cannot be overstated. They provide a core capability required to engineer large, complex systems where humans are considered to be candidate resources for performing system functions. It provides the capability to perform the critical intermediate analysis step in going from the ship mission-level requirements to ship HW/SW systems specifications.

Life-Cycle Cost Analysis—The requirements for CAIV have greatly impacted the systems engineering process. It is now required that total LCC and, as appropriate, TOC be the cost factor used in performing trade-off analysis. CAIV also requires that cost analysis must be performed earlier in the development process and with greater accuracy.

Because of the major impact of personnel on ship LCC, the first opportunity for reliable cost estimates is after an initial allocation has been made of humans to shipboard activities (operational process analysis). This corresponds to having identified some notion of the ship BMO, along with other major ship LCC drivers. However, the question becomes how to generate accurate cost estimates based on this notional ship concept.

The key in estimating ship costs before design is to use the structured operational process descriptions as a starting point. That is, operational costs are estimated first, based on the consumption of inputs (per unit time) by the individual activities of the ship operational process models.

Next, development or procurement costs of nonconsumable resources, such as humans and machines, are estimated based on their percent allocation to those activities. Machines and SW may require parametric-based cost estimating methods, especially for developmental items. The Navy's Cost of Manpower Estimating Tool provides the data (and roll-up methods) necessary for estimating the total cost of each manned position. Finally, maintenance costs are estimated based on the maintenance-level assignments of all allocated resources.

The primary methodology used in the BPR industry for cost estimation of combined HW, SW, and human systems is called ABC. In this methodology, activities are viewed as the reason that expenses are incurred. The ABC approach provides two separate means of generating cost savings. The first is through process streamlining, and the second is through more cost-effective resources allocation.

The strength of the ABC approach is that it can be used with activity (process) modeling for performing cost-performance analysis at the operational level. ABC is ideally suited for estimating operational costs of ship systems because shipboard operations (activities) need to become very well defined. That is, shipboard operations modeling must be done as part of ship functional analysis. Using ABC, operational costs would be estimated based on the quantity (per time) of consumable resources that are used (fuel, electricity, etc.) by all the activities.

Procurement costs can also be estimated using methods supported by an ABC capability. ABC provides methods for allocating nonconsumable resources (machines, personnel, space, etc.) to the process models. The descriptions of these nonconsumable components may take the form of parametric relationships between cost and relevant component attributes. The resulting ship operational model therefore ties component costs to operational processes. Most importantly, the operational process models are identically those used for assignment of performance metrics. Note that this kind of ship model would support very global cost-performance trade studies between ship operational processes, resources (machines, operators), and system component attributes.

However, ABC is probably not appropriate for analyzing offboard system maintenance costs. Maintenance, unlike shipboard operations, may involve activities that cannot be readily modeled. In addition, it is often difficult to verify the actual amount of human labor that has been historically "allocated" to maintenance tasks. ABC is most appropriate where it is necessary to define activities or model processes.

ABC supports the merging of cost estimating, systems engineering, and human engineering into a single comprehensive methodology. It utilizes similar types of modeling methods and constructs used for performing process and functional analysis. Cost estimating becomes an integral part of ship development. The resulting process-oriented methodology thus creates a consistent basis for conducting cost-performance trade studies.

Functional Analysis—Process and functional analysis of ship operations are needed to address system ownership cost issues such as maintainability and upgradability. However, functional analysis is also necessary in order to be able to specify the basic capabilities required of the ship.

Even though the performance of a system can be (theoretically) specified as a black box, functional analysis is needed to understand what those black box requirements are. That is, gaining an understanding of a requirement at any level of detail typically requires "drilling down" at least two more levels in detail. This rule of thumb used in functional analysis applies to specifications at the "performance" level as well.

Without formal functional analysis, functional partitioning of the system becomes haphazard. Worse yet, the resulting system functional partitioning will tend to reflect how the requirements were originally structured. High-level ship requirements are typically specified along warfare areas. Using simple decomposition of mission-level requirements will not identify common functionality across warfare areas or their associated systems.

Functional analysis needs to be done regardless of whether the system will be eventually modeled as objects. Object definition can be viewed as an initial design activity, whereas functional modeling is a (last-step) analysis activity. Object definition involves combining functions (object methods) and data (object characteristics) into system objects. It thus represents more of a synthesis activity. That is, the decisions made regarding object definitions begin to define the high-level system design (system architecture).

Prototyping & Trade Studies

The TSSE model of Figure 1 brings to the forefront the importance of conducting trade studies in defining ship requirements. Trade studies are conducted to trade-off ship systems performance, cost, and risk factors. The trade studies include mock-ups and prototypes to identify and assign specific performance metrics. Trade studies are typically required at all levels of requirements derivation (mission, operational, and architecture) in the analysis of system performance and cost metrics.

The requirements and functional analysis models help identify and define the critical trade studies or experiments to be conducted. The analysis models are also used to capture the experimental results and translate them into performance metrics. That is, the experimental results from the trade studies refine the models and, thus, refine the requirements. In addition, prototypes that have been developed as part of conducting trade studies can be used to verify the (functional) analysis and associated simulation models.

Utilizing analysis/modeling in conjunction with prototyping can address many of the problems traditionally associated with prototyping. These problems have typically included a lack of well-defined goals or metrics in setting up the experiments. This, in turn, results in another problem typically associated with prototyping: "lessons learned" not being fully captured. Analysis and modeling can be used to provide structure for both setting up the experiments and capturing the results.

Prototyping and mock-ups serve multiple purposes. First, they can help identify what the Navy wants and needs. That is, prototyping provides one of the best means for requirements validation. Second, prototyping used in conducting trade studies can provide detailed information on system cost versus performance relationships. Third, prototyping educates the customer on various aspects of system design. Customer education is key for effective contractor proposal evaluations and technical reviews. In fact, prototyping can often provide important lessons to both government and industry teams. Lastly, prototyping is one of the primary means of reducing technical risk.

Prototyping and Risk Reduction

Prototyping is the primary means whereby technological risks are reduced. Prototyping provides familiarity and understanding of candidate technologies and components. It is an important customer-sided risk reduction activity needed to support requirements definition.

Anything new or unknown represents risk. Although the technologies used in COTS may not be new, the use of COTS components does not necessarily reduce risk for the customer. Venders may use proven technologies in order to reduce risk for themselves. However, this does not necessarily translate into reduced risk for the buyer. The reason for this is that there is typically much about the component that does not get revealed to the buyer, in both functionality and performance.

Although vendors may provide technical information on their products, this information is often incorrect or applies only to a very narrow application of their product. Purchasing components that have not been verified by the customer for their particular application is very risky. In addition, there are future risks associated with a lack of control over product changes. This is especially true for defense systems that often have consequential safety considerations. Verification at the black-box level can be difficult and expensive. However, prototyping and testing provide the best means for reducing the risks associated with the use of COTS.

Prototype Transitioning

Prototyping can also provide a source of components and modules to transition onto the ship. The key to transitioning of prototypes it is to maintain a flexible view of the prototype itself. When a prototype is viewed only in terms of a product, this very much limits transition possibilities. Assuming that the system is modular in design, it is typically advantageous for transitioning to view the system as individual components or modules.

Typically, smaller modules can be more readily integrated into a new system without corrupting its architecture. In addition, larger modules tend to have unneeded functionality. For each prototype module, a decision must be made to either use, discard, or reengineer.

A risk assessment should also be performed on each module based on the amount and quality of information available. Smaller modules are typically more easily tested and/or come with better test information. For large modules, the assessments themselves can be risky because of the time and effort that may be required.

Integration

Allowing for requirements refinement during early ship design can have the additional benefit of shortening development time. However, this can significantly increase the amount of concurrency in the development process. That is, development activities that were performed sequentially (waterfall process) may become concurrent.

Concurrency implies the need for much greater coordination. This, in turn, implies the need for a more integrated development environment. Historically, requirement changes during development imposed great strains on large programs. There were no easy or automated means of estimating their full impacts. The customer would of course tend to underestimate the impacts, and the builder would tend to overestimate them.

Greater concurrency or parallelism in a development program places additional integration requirements on the supporting tools and data management environment (see Integration in Figure 1). This infrastructure must support data and process integration of the analysis, trade studies, and design activities. It must support a development process where changes to both requirements and design are habitual.

Engineering Tools Environment

A distributed, integrated engineering tools environment is sometimes referred to as an "enterprise." These environments typically are composed of engineering tools that are integrated by a communications and data management infrastructure. A desirable feature of an engineering environment is to have both the analysis and design tools integrated. An engineering environment should also have established processes for performing engineering activities, with a focus on engineering tools use. This is because the tools themselves incorporate methods that need to be meshed with the project process definitions.

The process definitions of the integrated engineering environment should be reflective of an overall operational concept for the engineering environment. In fact, the operational concept can be viewed as establishing the requirements for the tools, methods, and their integration. An operational concept for the development environment can also be the basis for establishing an integrated set of program/project policies and procedures.

The environment should strongly support traceability between project elements. That is, all aspects of the project and system from requirements to implementation should be linked by well-defined relationships. Changes made to any of the project or system elements should initiate a response from the environment indicating those other elements that may need to be updated.

The environment should also be capable of capturing various types of analysis and trade studies results. This requires that system elements can be assigned cost, performance, and risk factors or attributes. The environment should also support decomposition and roll-up of these factors.

Another key aspect of the engineering environment is that it must aid in maintaining consistency (along with correctness) between system elements and their attributes. A necessary condition for this is that there is but one integrated, or at least a single coordinated, data management infrastructure that supports development. Having a single comprehensive infrastructure is an effective way to ensure that cost, performance, and risk factors are attributed to the correct and appropriately defined ship system element.

Ideally, the system analysis and design tools should be integrated through a global data management system. This system should have a schema (database design) that can be readily adapted to accommodate new tools and their capabilities. The schema should be capable of evolving to reflect the structure of the ship design and the analysis models.

Program Data Management Environment

Many Navy system development programs of today are attempting to leverage large amounts of legacy system HW and SW. Requirements are being levied on existing systems to support new mission areas. In fact, these development programs may better be characterized as large reengineering projects.

As an additional complication, the leveraged systems are typically continuing to add capabilities to better support their current mission functions. That is, the individual systems that are to be leveraged have configuration baselines that are independent of their new mission requirements. Thus, there is much need for an integrated *Program Data Management Environment* that spans across individual programs. This is absolutely necessary for tracking the development of component systems so that new composite capability can be planned for and implemented. It is also a necessary condition for eventual programs and systems integration.

The program-level data (schedules, milestones) of the individual systems must also flow down to the project or systems level. This is necessary to ensure that the requirements for new capabilities are consistent with the development schedules of the individual programs. That is, it is important that the *Program Data Management Environment* is coupled to the *Engineering Tools Environment* (see Figure 1) through the system requirements. This allows for tracking the capabilities of the individual systems so that development of new capabilities can be planned.

Certainly a program Work Breakdown Structure (WBS) for ship development needs to adapt according to the different types of efforts required in the various program phases. However, this linking of program and systems data would also support an evolution of program WBS according to ship definition. That is, it would allow for the program WBS to more readily evolve in order to reflect refinements to the ship functional decomposition. Without this, programs tend to remain "stovepiped" according to the original mission-oriented structuring of the requirements.

CONCLUSION

Systems engineering needs to become an integral part of the program management decision-making process. Informed decision-making via systems analysis is critical for program control. Systems analysis at all levels of detail must continue throughout development to support effective monitoring and control of contractor efforts.

Today there are process and functional analysis tools that can be used for both understanding and specifying complex systems without getting into design details. Ship operational process modeling is key in transitioning from mission-level requirements to system specifications. Prototyping, when used in conjunction with these analysis methods, establishes a powerful trade-studies capability for specifying value systems. With today's emphasis on COTS use, prototyping also becomes an important customer-sided, risk-reduction activity.

Customer-sided analysis efforts need to be coupled to builder-sided design efforts. This implies the need for an integrated development environment for both government and our industry partners. Just as important is the need for an integrated program data-management environment. This represents a core capability for enhancing existing systems to support new mission areas. Indeed, a more capable and integrated development infrastructure is the basic prerequisite for TSSE.

BIBLIOGRAPHY

Clare, T.C. and Holden, R.A., "Engineering 21st Century Systems," *Naval Engineers Journal*, Sep 1996.

Clark, John and Lamb, Thomas, "Build Strategy Development," *Journal of Ship Production*, Aug 1996.

Duren, B.G.; Goldstein, S.; Chima, S.; Embree, D.; Keane, R. and Pollard, J.R., "NSWC Total Ship Teaming Initiative: A New Front End Process for Ships and Systems," *Proceedings of the ASNE Symposium on Engineering the Total Ship*, May 1998.

Duren, B.G. and Pollard, J.R., "Building Ships as a System: An Approach to Total Ship Integration," *Naval Engineers Journal*, Sep 1997.

Gray, R.E. and McCollum, K.G., "Risk and Focus in Defense Systems Acquisition," *Proceedings of the ASNE Symposium on Engineering the Total Ship*, May 1998.

Hockberger, W.A., "Total Ship System Design in a Supersystem Framework," *Naval Engineers Journal*, May 1996.

Jons, Otto; Ryan, J. Christopher and Jones, Gary W., "Using Virtual Environments in the Design of Ships," *Naval Engineers Journal*, May 1994.

Koenig, P.C. et al., "Design Infrastructure in Shipbuilding and Other Heavy Industries," 1996 Annual Meeting SNAME, NY.

NAVSO P-6071, *Best Practices: How to Avoid Surprises in the World's Most Complicated Technical Process*; Mar 1986.

Rains, Dean A., "A System Engineering Approach to Surface Combatant Design Issues," *Naval Engineers Journal*, May 1990.

Report of the Total Ship Engineering Working Group, prepared jointly by SEA-05 and SEA-06, Naval Sea Systems Command, 19 Aug 1988.

Scacchi, Walt and Boehm, Barry, "Virtual System Acquisition: Approach and Transitions," *Acquisition Review Quarterly*, Spring 1998.

THE AUTHOR

MR. JAMES D. HORNER

Mr. James D. Horner received B.S.E.E. and M.S.E.E. degrees in computer engineering and an M.B.A. in information systems from University of Maryland in May 1983, December 1986, and May 1991, respectively. From 1991 to 1994, he served as Deputy Director of ASW Technology Programs at the Naval Surface Warfare Center, Dahlgren Division, White Oak. Since then, Mr. Horner has been providing systems engineering expertise for the transitioning of combat system technologies and related development processes for various Navy technology programs. He is a member of the Fredericksburg Chapter of the International Council on Systems Engineering.